CSI4107 Final System Report

Xiaoxin Zhou

7957115

Modules Implementations and Descriptions

Module 8a - Bigram Language Model

In this module, I created a function called `bigram_model(list_of_dicts)` which results in a bigram model. The input is a list of dictionaries, in which each dictionary represents one document, or one news. Each dictionary contains keys `topic`, `description` and `name`, I only used `description` information when I was building the bigram model. First, I extracted this information from each dictionary and concatenated them into one list. Second, split the strings and got a list of words. Then I created a finder by using the `BigramCollocationFinder` that from `nltk.collocations`. This finder has a function `apply_freq_filter(10)`, the number can be determined by the users. This function is used to remove all bigrams which have frequencies less than 10 because they might not be a relevant bigram of the language. In my model, I chose the filter number is 5. In addition, sort the bigram lists in descending order of their frequencies.

Since the size of the data is quite big, for the convenience of the future usage, I saved the result data into a numpy array file, named `bigram_model.npy`. If you want to see the structure of the data, use `res = np.load('bigram_model.npy')`, in which `res` is a numpy array. `res[0]` returns `array([[(('mln', 'dlr'), 9109)]])`.

Module 8b - Query Completion Module

This module connects the user interface with the bigram model to allow users input a query and see the suggestions given by the bigram model. When I was building this module, I searched online and found a clearly-explained tutorial from w3school

The given example in the tutorial is written in JavaScript, and the suggestion terms are all put in a list. In another word, the dictionary of terms is manually defined. In my case, when user input a word, the system searches in the bigram model, and return the suggestions based on the values they have. For example, if user type word `apple`, there are two choices in the bigram model: `[('apple','fruits',900)]` and `[('apple','phone',30)]` The system should return `fruits` as suggestions because 900 is bigger than 30. I tried programmed this procedure in Python.

My challenge here is the connection between JavaScript and Python. In particular, how to pass the result of a Python function to JavaScript.

Module 9a - Automatic Thesaurus Construction

When I was constructing the thesaurus, I firstly design the structure first. There are two similarity measures given in the instruction: Jaccard, and Cosine. I chose Jaccard because it is easily calculated and it returns a symmetric matrix, which occupies less space.

Before I constructing the thesaurus, I already have a term-frequency matrix, which will be introduced in the later in this report. In this term-frequency matrix, columns are around one-thousand terms, and each row represents a document. The value in each cell stands for the number of occurrences that the term occurs in its corresponding document.

The goal in this step is to create a new matrix which both rows and columns are the terms, the value in the cell is the similarity calculated from Jaccard method. Since the size of matrix would be huge, and the matrix is symmetric, I calculated only half of the matrix, which contains all the similarities among all terms. Firstly, I created a dictionary, in which a key is a term, the value is a list of documents that the term occurrs. Secondly, compare the lists between two lists of two terms to calculate the similarities. My approach to shorten the running time is every comparison on term *n* starts from *n+1*. For example, there are 5 terms: *t1, t2, t3, t4, and t5*, after comparing (*t1, t2*), there is no need to calculate (*t2, t1*) again, just starts from (*t2, t3*).

In the end, after I calculated all measures, I saved the result in the `thesaurus.csv`.

Module 9b - Global Query Expansion (in VSM)

Based on the thesaurus built in the previous module, the similarities between each term can be used to expand the query. When user input a word, the system searches the term in the thesaurus. I assumed the filter rate is 0.75, which means, only the terms have a similarity greater than 0.75 will be returned in the expansion. If user query contains more than one word, I chose the first and the last term to be used in the query expansion.

Module 10a - Text categorization with KNN

I did the topic assignment at very beginning in this project, only after the dictionary construction. Because the whole dictionary contains both the course data and the reuters data, I only need the reuters data to assign the topics, I created a dictionary only contains reuters data. When I was parsing the documents in the process of dictionary building, I filled *Nan* in each null values, such as topics, descriptions, and titles.

In the procudure of KNN, firstly I separated the files into two parts: traning data: `train_df` and the predicting data: `predict_df`. Next, I handled the cases which one document contains many topics. The function called `manyTopics(df)`. My approach can be illustrated as follows:

Before:

index	Document	Topic(s)
1	doc1	oil, weather, education
2	doc2	movie, music

After:

index	Document	Topic(s)
1	doc1	oil
2	doc1	weather
3	doc1	education
4	doc2	movie
5	doc2	music

The second step is cleaning the training data. In particular, I combined the description and name, transform all letters into lower cases, remove the punctuations, numbers and some special characters such as '\x'. Moreover, I preprocessed the training data, the function I used is `preprocessing(string)`, which in the Vanilla System. When I finished these processes, the `train_df` looks like this:

index	Document ID	Topic	Text
30	1	oil	[econom, data, thi, week, could, ...]
31	1	weather	[econom, data, thi, week, could, ...]
15	2	education	[shr, loss, dlr, net, ...]

By following the idea given in the lecture, I need to create a vector space for the most common terms. Thus I concatenated all list of terms in the `text` column, and calculated the terms that occurrs more than 100 times by using `Counter` from library `collections`. This step reduces the size of data and finds the most popular terms.

Next, I used these selected top-occurrance words to create the vector space, then insert the training data into this vector space. The functions used here are `createVS(topTerms)` and `insertVS(sourceDF)`.

In the modeling step, I used `KNeighborsClassifier` from library `sklearn.neighbors` to fit the model. At the same time, I proprocessed the `predict_df` to get the same format as `train_df`. Then I loop over the `predict_df` , predict and asign the topic for each document.

Module 10b - Topic Restriction

After filled topic to each document in the `predict_df`, combined the train and predict data together. Then I created a dictionary which keys are the topics, values are the lists of documents.

The part of the dictionary looks like : `{'interest':[4, 89, 130]}`. In the user interface, when user input `interest` at the topic box, the results are document with id 4, 89 and 130.

Deal with Reuters collection

I used the library `glob` to read all files with the `*.sgm` extension. Then I created a function `build_dict(doc)` that parses a single `xxx.sgm` file. In the parsing process, I used `BeautifulSoup` to parse tags. The most high-level tag is `<reuters>`, which separates each document. Under each document, I extracted topic, description, and title information and saved them in to lists in function `` `parseFile(reuters) ``. The corresponding tags are `<topics>, <body>,` and `<title>`. Missing values need to be handled here by filling `'Nan'`. For the `<topic>` tag, the document with multiple topics have multiple child tag `<d>`, so I created a list to save all topics.

In the function `rt_corpus_preprocessing(docs)`, I looped over all `.sgm` files and assigned `docID` to each document. For convenience, I saved the final dictionary into a Json file called `reuter_dic_lst.json` , I did the same for the course data: `course_dic_lst.json`. One advantage is I don't have to re-run the whole function every time. Another try I did in this project is I programmed the project on the Jupyter Notebook, which allows me to try and test on very small piece of data first and then apply to the actual data. It saved me a lot of time than programming on the normal editor, and Jupyter Notebook displays pandas dataframe elegantly.

In the whole project, the most time-consuming part is the thesaurus construction. Compare with this costruction, the dictionary building doesn't consume a lot of time, especially I saved the parsed data into the json file.

Compared with the CSI collection data, the reuters collection data has a more complex file structure. I took sometimes to figure out how to extract exact information from different tags. Another challenge is the size of the data, since the reuters collection data has a huge size, and I always try and test when I was programming. Thus, every time I want to do some modifications, I saved the original data and created a copy of the data, do the modifications on the copied data first. This way helps me to avoid some big mistakes that needs to re-run the program from the beginning. But when I was doing some housekeeping work, such as put lines of code into a function and migrate functions to the Python file, I might made mistakes such as the inconsistency of variable names.

Test queries:

shareholder AND security

Results

japan plans more flexible corporate bond issues

readMore

canada sets off its own big bang

readMore

pesch sees shareholder support in ami ami bid

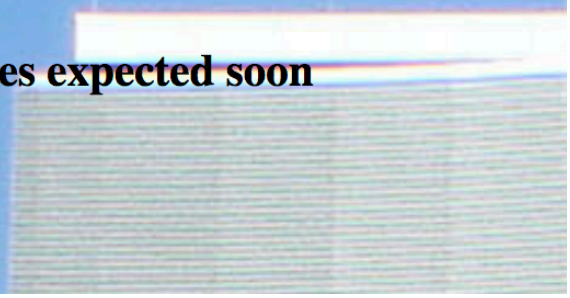
readMore

pefco sells notes via jp morgan securities

readMore

global trading in yen bond futures expected soon

readMore



canada sets off its own big bang

stock market deregulation will be heard tuesday when the canadian securities industry sets off its own fireworks some companies are quickly taking advantage of deregulation but more significant moves are likely to be a year or more down the road analysts say they say most foreign firms will focus at first on bond trading debt financing and underwriting areas in which they have experience under the old rules that allowed them to provide government debt financing and advisory services when they learn about canada they might go into retail said carney at merrill lynch canada then they will eventually just start to buy up the little guys if the bigger boys price comes down they ll end up taking those ones over too under the new rules barriers that kept banks brokerage houses insurance companies and trust firms out of each others businesses will be swept away in favor of wide open competition the deregulation sometimes nicknamed little bang will allow canadian banks and other domestic financial institutions to set up securities units or acquire existing brokerages foreign companies will also be able to form their own securities divisions in ontario or buy up to pct of a canadian firm a year from now they can raise that stake to pct foreign firms that entered canada after can currently own no more than pct of a canadian brokerage deregulation was expected to generate a major shakeout by june involving mergers of existing firms and buyouts by foreign and domestic financial giants so far few deals have become reality however despite a swirl of rumors and merger talks in the heat of deregulation fever many canadian investment dealers are demanding up to four times book value scaring off prospective suitors analysts said i would think they the brokerages will have to lower their asking prices said financial services analyst jeff carney at merrill lynch canada inc which recently terminated talks to acquire prominent canadian brokerage burns fry ltd i think a high asking price is what scared merrill lynch away from burns fry added carney it is just a lot of money when your asset walks out the door every night it is difficult to pay that amount of cash carney said another factor in the slow pace of buyouts was uncertainty about the final deregulation rules which were issued and approved only in mid june most major foreign and canadian financial players have focused their deregulation strategy on planning their own canadian securities units or buying seats on the toronto stock exchange canada s largest equities market japan s nomura securities co ltd yamaichi securities co ltd daiwa securities co ltd and nikko

oil AND profit

Results

energyheavy oils

readMore

energyforeign investors

readMore

division seen on how to help us oil industry

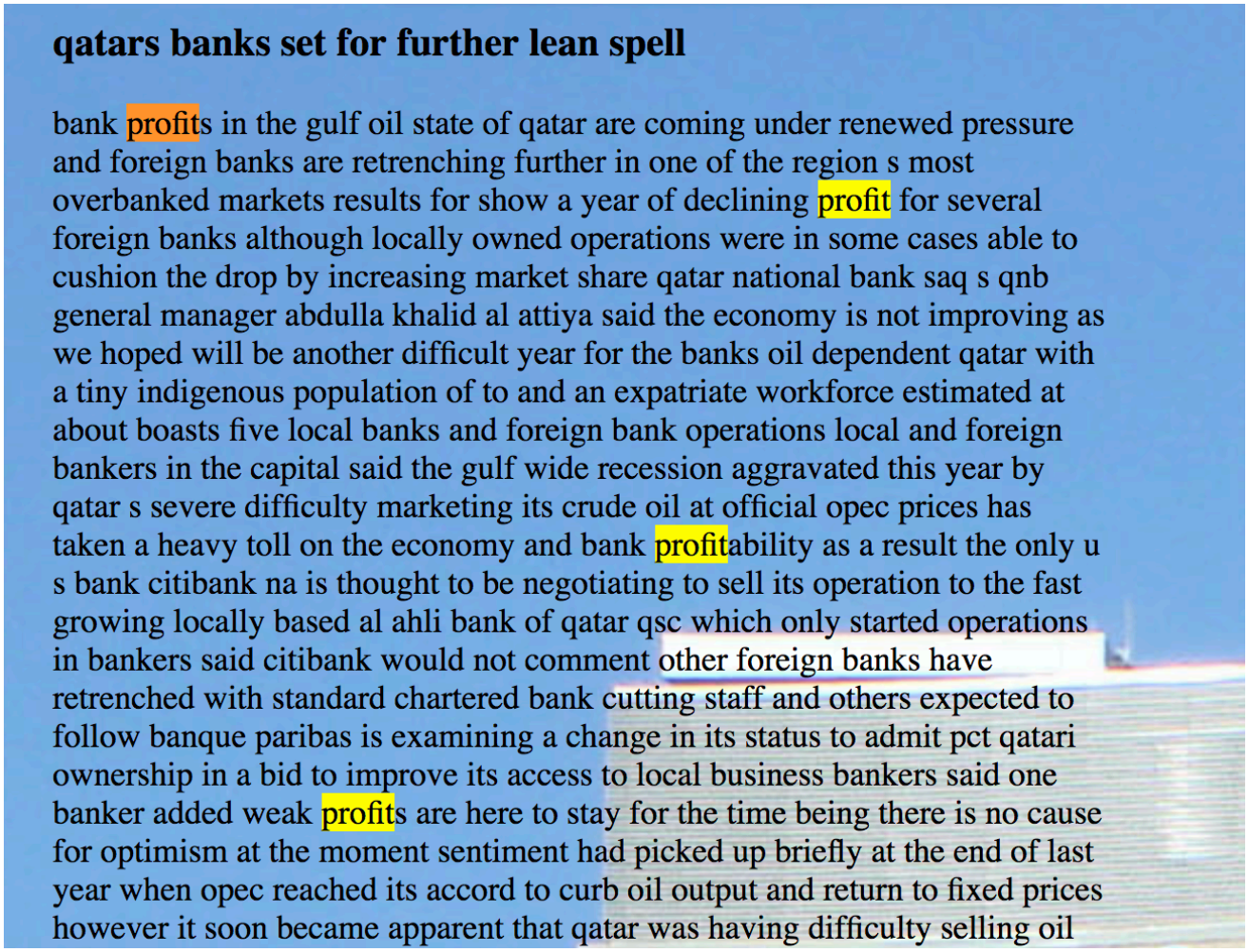
readMore

qatars banks set for further lean spell

readMore

canada oil industry set for recovery analysts

readMore



Canada canola oil

Results

energyheavy oils

readMore

canada oil industry set for recovery analysts

readMore

energyforeign investors

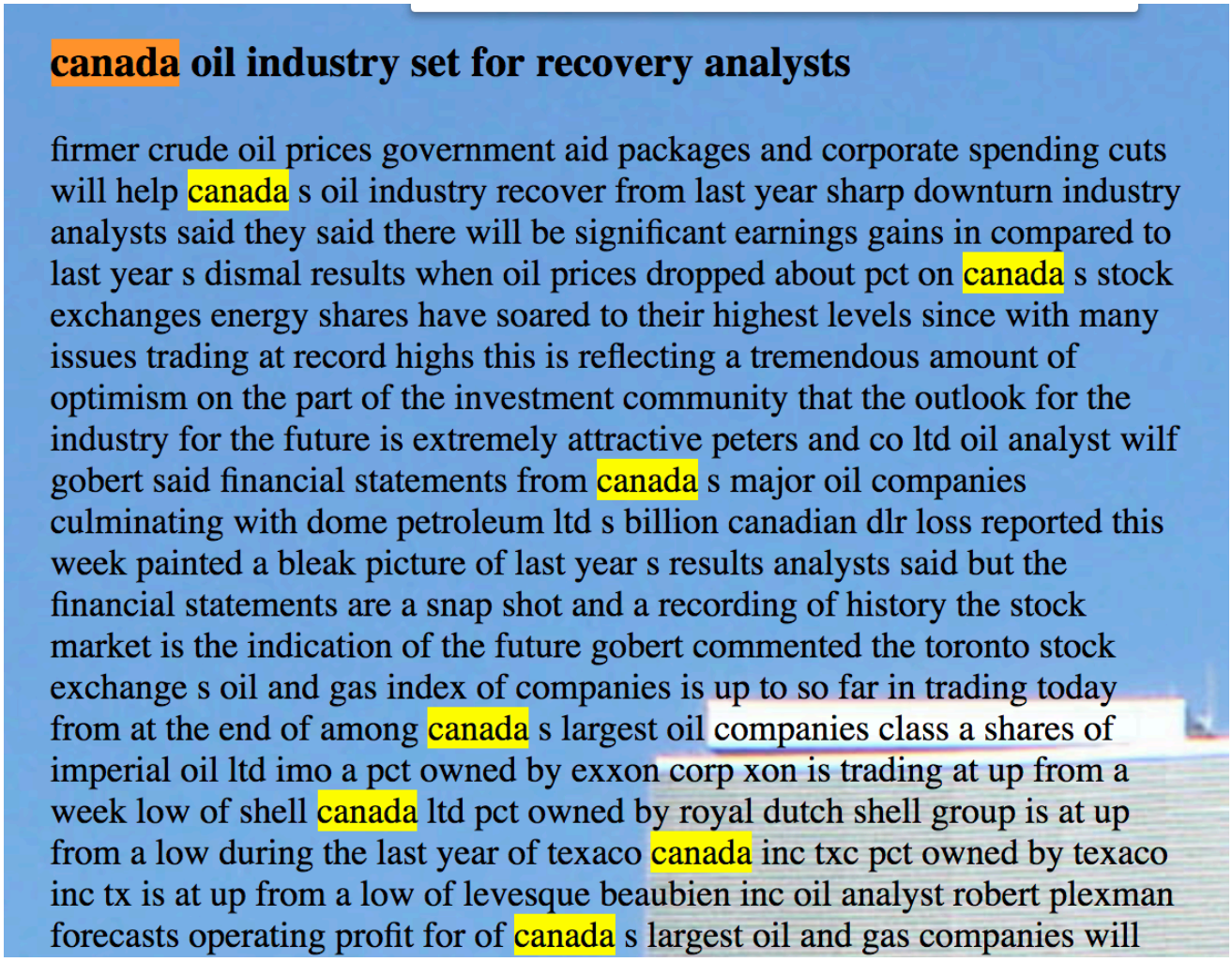
readMore

petrocanada to drill two terra nova test wells

readMore

division seen on how to help us oil industry

readMore



European banks stockholders

Results

japanese banks expand hong kong presence

readMore

japanese banks expand hong kong presence

readMore

canada banks could see pressure on brazil loans

readMore

british banks resist signing mexico package

readMore

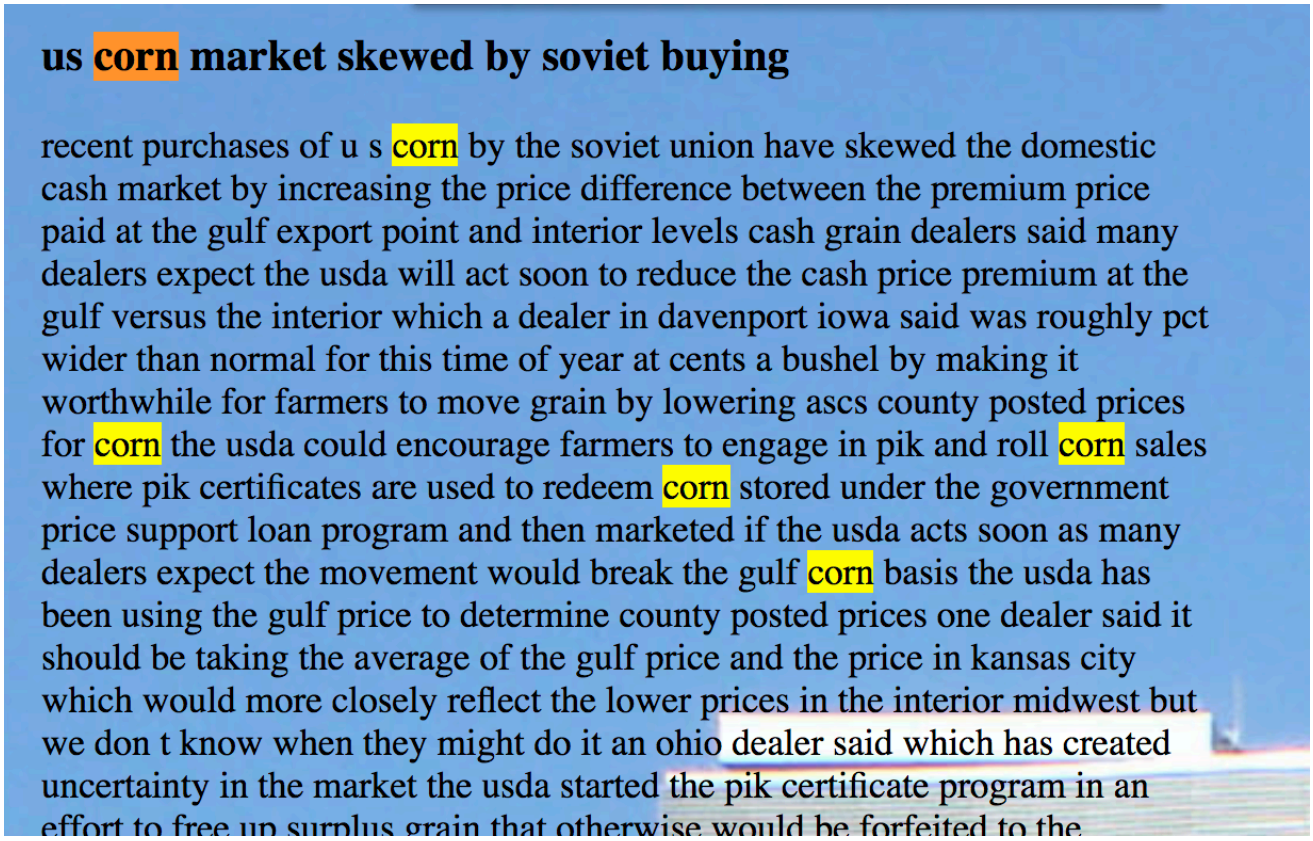
bank of england proposes limits on underwriters

readMore

bank of england proposes limits on underwriters

the bank of england proposed that banks in britain for the first time will have to seek authorization to be lead underwriters of new issues in the euromarkets in addition the bank has proposed limits on the amount of credit risk that firms can take on with each new issue they lead manage the proposed rules were outlined today in a consultative paper on large credit exposures taken by recognized banks and deposit takers and may be modified the bank said the rules could affect billions of dollars of eurobond securities issued each year in about billion dlrs of new eurobonds were offered according to figures compiled by euromoney magazine up until now there have been no restrictions on lead managing eurobond issues however eurobond market participants believe the bank of england is concerned about intense competition for market share which may have caused some firms to expose themselves to imprudent credit risk competition for market share has been a

U.S. corn market



Query Expansion

```
▶ py spider.py
previous documents found loading into memory
================================================================================================================================
[coffee] being expanded to ['membership', 'water', 'concess', 'burlington', 'opinion', 'bar', 'grant', 'calcul', 'feel', 'front']
================================================================================================================================
================================================================================================================================
[stock] being expanded to ['ago', 'tradit', 'imf', 'street', 'ongpin', 'grew', 'ministri', 'recov', 'temporari', 'yield']
================================================================================================================================
================================================================================================================================
[oil] being expanded to ['provision', 'extraordinari', 'independ', 'disclos', 'encourag', 'ge', 'distribut', 'analyst', 'compromis', 'made']
================================================================================================================================
================================================================================================================================
[course] being expanded to ['risen', 'shortli', 'feed', 'becam', 'borg', 'broad', 'pc', 'smaller', 'significantli', 'wright']
================================================================================================================================
================================================================================================================================
[corn] being expanded to ['complaint', 'compens', 'advertis', 'especi', 'smith', 'bodi', 'ordinari', 'receipt', 'iraqi', 'jet']
================================================================================================================================
```

Topic Classification

The k I chose is the square root of the number of rows. The details of topic assignment in the above description.